ASSIGNMENT 8

September 15, 2021

1 ASSIGNMENT 8

2 SOUMYA GITE

3 SIRSS2276

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib as mpl
%matplotlib inline
```

[2]:
```
train_data = pd.read_csv('heart.csv')
test_data = pd.read_csv('o2Saturation.csv')
```

[3]:
```
train_data.head()
```

[3]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	\
0	40	M	ATA	140	289	0	Normal	172	
1	49	F	NAP	160	180	0	Normal	156	
2	37	M	ATA	130	283	0	ST	98	
3	48	F	ASY	138	214	0	Normal	108	
4	54	M	NAP	150	195	0	Normal	122	

	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	N	0.0	Up	0
1	N	1.0	Flat	1
2	N	0.0	Up	0
3	Y	1.5	Flat	1
4	N	0.0	Up	0

[4]:
```
test_data.head()
```

[4]:

	98.6
0	98.6
1	98.6
2	98.6

```
3  98.1
4  97.5
```

[5]:
```
train_data.columns
```

[5]:
```
Index(['Age', 'Sex', 'ChestPainType', 'RestingBP', 'Cholesterol', 'FastingBS',
       'RestingECG', 'MaxHR', 'ExerciseAngina', 'Oldpeak', 'ST_Slope',
       'HeartDisease'],
      dtype='object')
```

[6]:
```
train_data.shape
```

[6]:
```
(918, 12)
```

[7]:
```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Age             918 non-null    int64  
 1   Sex             918 non-null    object 
 2   ChestPainType   918 non-null    object 
 3   RestingBP       918 non-null    int64  
 4   Cholesterol     918 non-null    int64  
 5   FastingBS       918 non-null    int64  
 6   RestingECG      918 non-null    object 
 7   MaxHR           918 non-null    int64  
 8   ExerciseAngina  918 non-null    object 
 9   Oldpeak         918 non-null    float64
 10  ST_Slope        918 non-null    object 
 11  HeartDisease    918 non-null    int64  
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

[8]:
```
test_data.shape
```

[8]:
```
(3585, 1)
```

[9]:
```
test_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3585 entries, 0 to 3584
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   98.6    3585 non-null   float64
```

```
dtypes: float64(1)
memory usage: 28.1 KB
```

[10]:
```
train_data.describe()
```

[10]:
```
                Age   RestingBP  Cholesterol   FastingBS       MaxHR  \
count  918.000000  918.000000   918.000000  918.000000  918.000000
mean    53.510893  132.396514   198.799564    0.233115  136.809368
std      9.432617   18.514154   109.384145    0.423046   25.460334
min     28.000000    0.000000     0.000000    0.000000   60.000000
25%     47.000000  120.000000   173.250000    0.000000  120.000000
50%     54.000000  130.000000   223.000000    0.000000  138.000000
75%     60.000000  140.000000   267.000000    0.000000  156.000000
max     77.000000  200.000000   603.000000    1.000000  202.000000

          Oldpeak  HeartDisease
count  918.000000    918.000000
mean     0.887364      0.553377
std      1.066570      0.497414
min     -2.600000      0.000000
25%      0.000000      0.000000
50%      0.600000      1.000000
75%      1.500000      1.000000
max      6.200000      1.000000
```

[11]:
```
train_data.isnull().sum()
```

[11]:
```
Age               0
Sex               0
ChestPainType     0
RestingBP         0
Cholesterol       0
FastingBS         0
RestingECG        0
MaxHR             0
ExerciseAngina    0
Oldpeak           0
ST_Slope          0
HeartDisease      0
dtype: int64
```

[12]:
```
train_data.duplicated().sum()
```

[12]:
```
0
```

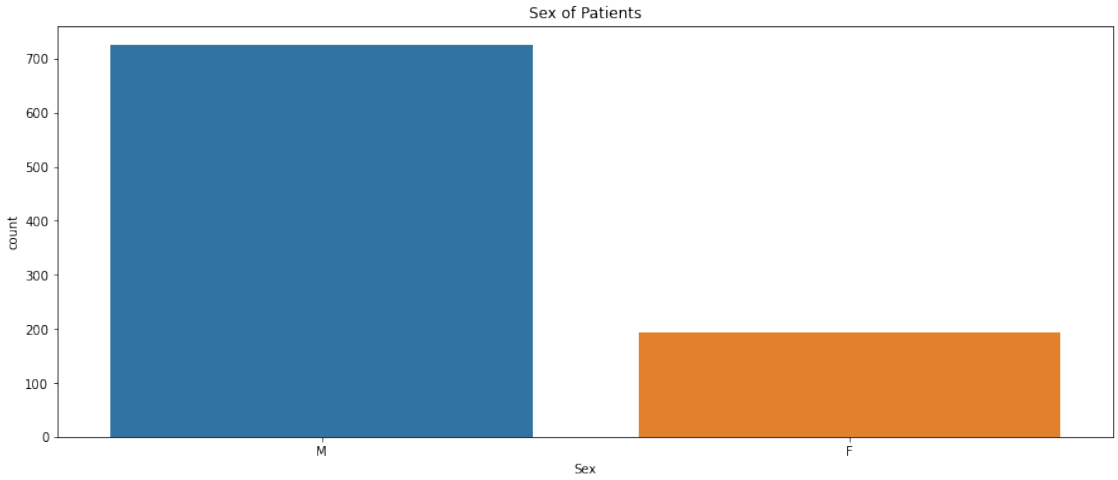
[13]:
```
test_data.isnull().sum()
```

[13]:
```
98.6    0
dtype: int64
```

[14]:
```
train_data.drop_duplicates(inplace=True)
```

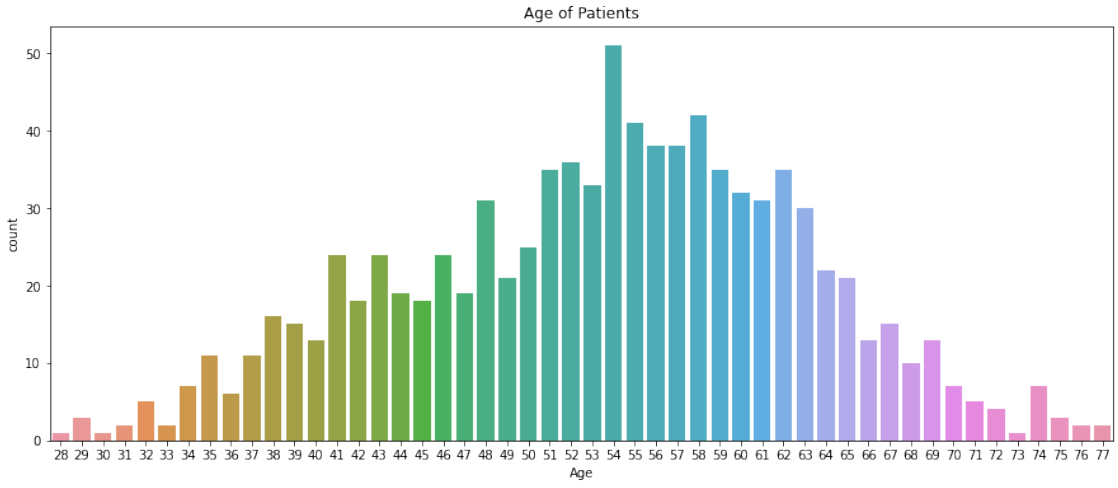
[17]:
```
plt.figure(figsize = (15,6))
sns.countplot(x="Sex",data=train_data)
plt.title('Sex of Patients')
```

[17]:
```
Text(0.5, 1.0, 'Sex of Patients')
```

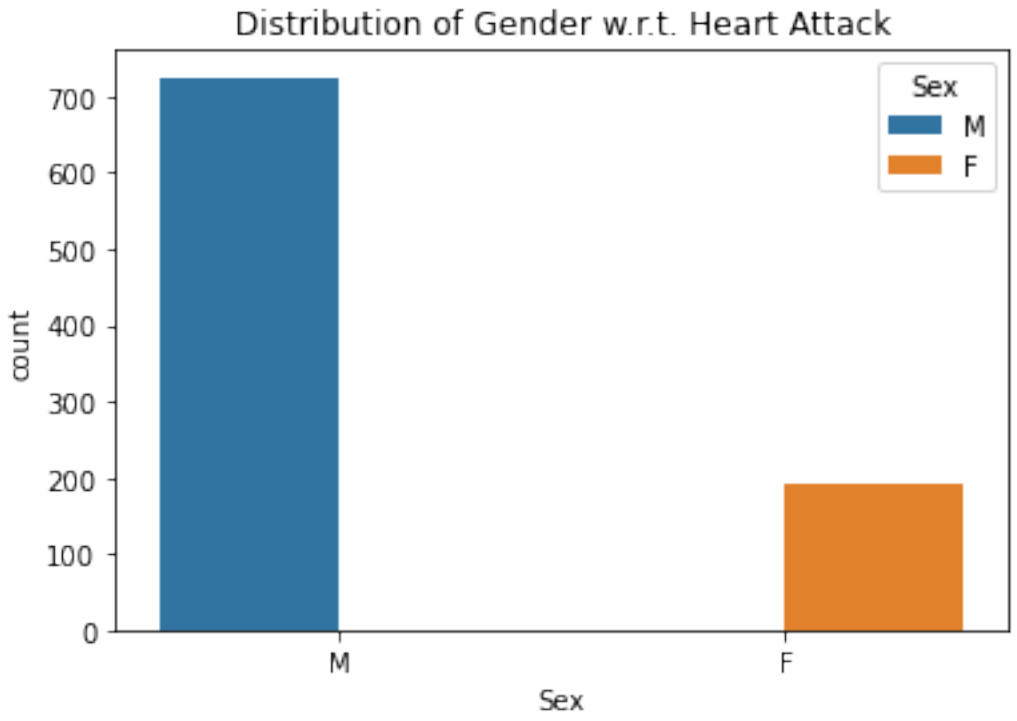


[18]:
```
plt.figure(figsize = (15,6))
sns.countplot(x="Age",data=train_data)
plt.title('Age of Patients')
```

[18]:
```
Text(0.5, 1.0, 'Age of Patients')
```

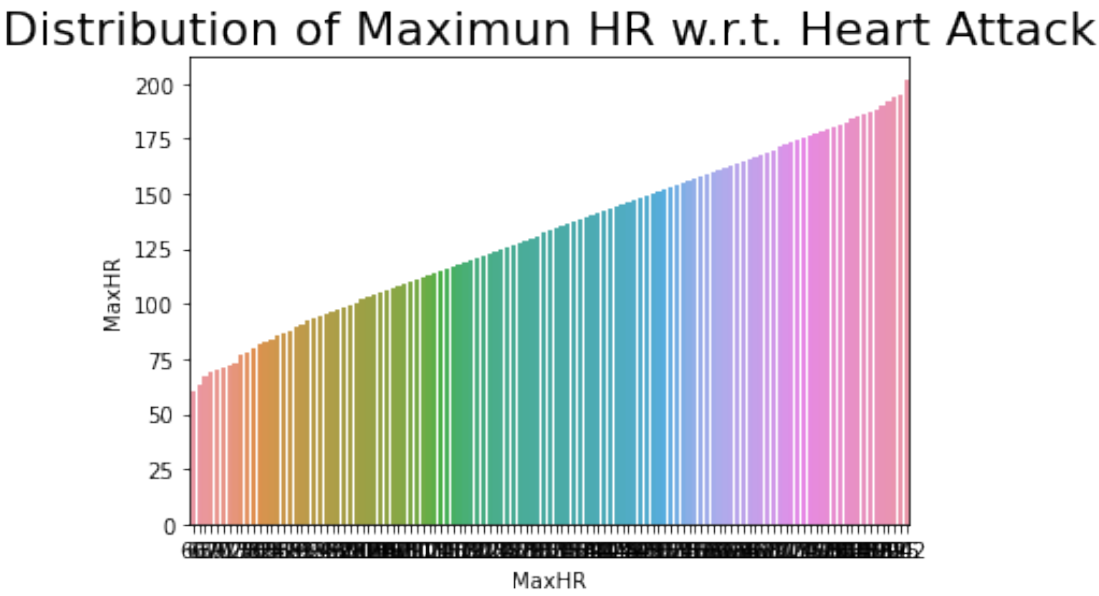


[50]:
```
sns.countplot(x = 'Sex' , hue = 'Sex' ,  data = train_data)
plt.title("Distribution of Gender w.r.t. Heart Attack")
plt.show()
```



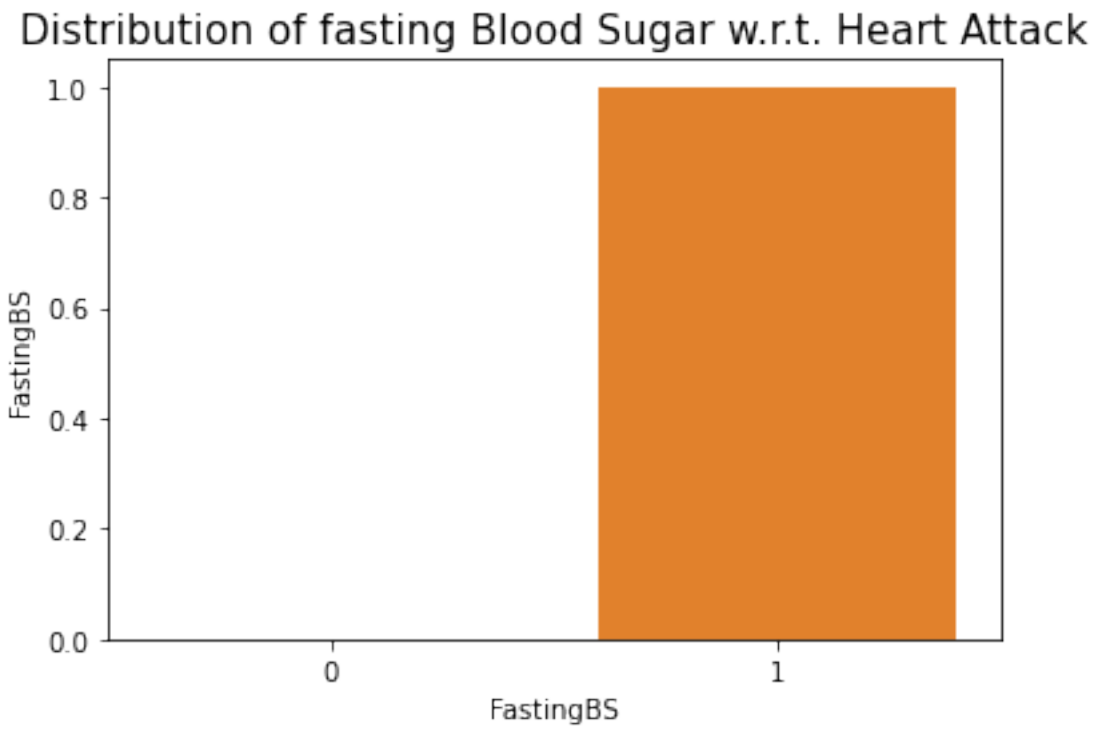
[68]:
```
sns.barplot(x = "MaxHR",y ='MaxHR' ,data=train_data);
plt.title("Distribution of Maximun HR w.r.t. Heart Attack" , fontsize= 22)
```

[68]:
```
Text(0.5, 1.0, 'Distribution of Maximun HR w.r.t. Heart Attack')
```



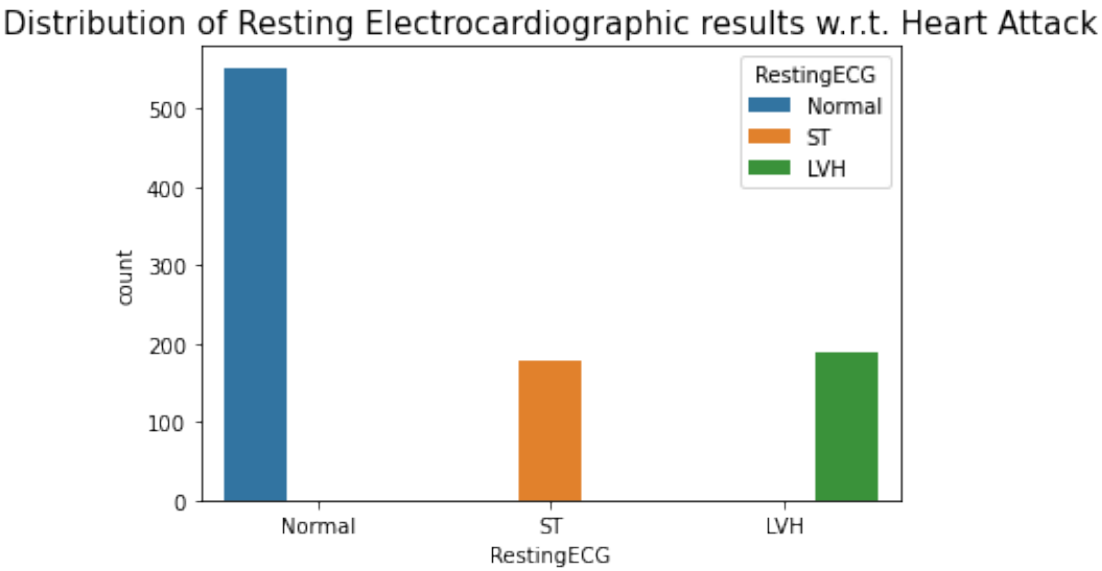
[60]:
```
sns.barplot(x="FastingBS",y ='FastingBS' ,data=train_data);
plt.title("Distribution of fasting Blood Sugar w.r.t. Heart Attack" , fontsize=␣
↪15)
```

[60]: Text(0.5, 1.0, 'Distribution of fasting Blood Sugar w.r.t. Heart Attack')



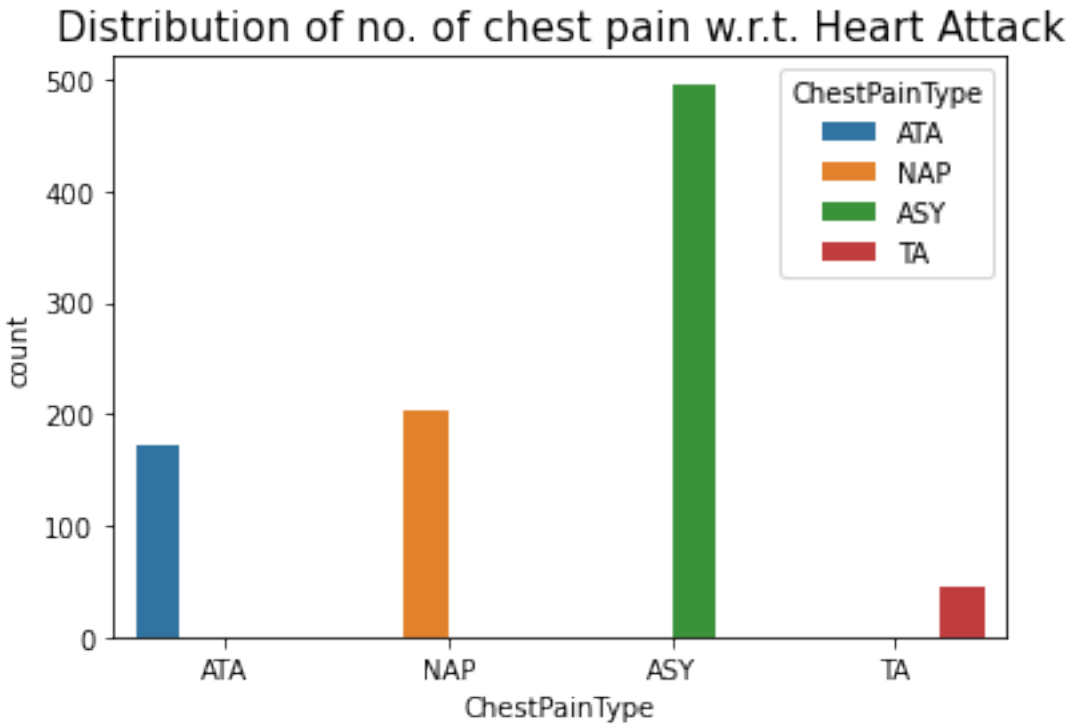
[64]:
```
sns.countplot(x = 'RestingECG' , hue = 'RestingECG' ,  data = train_data)
plt.title("Distribution of Resting Electrocardiographic results w.r.t. Heart␣
 ↪Attack" , fontsize= 15)
```

[64]:
```
Text(0.5, 1.0, 'Distribution of Resting Electrocardiographic results w.r.t.
Heart Attack')
```



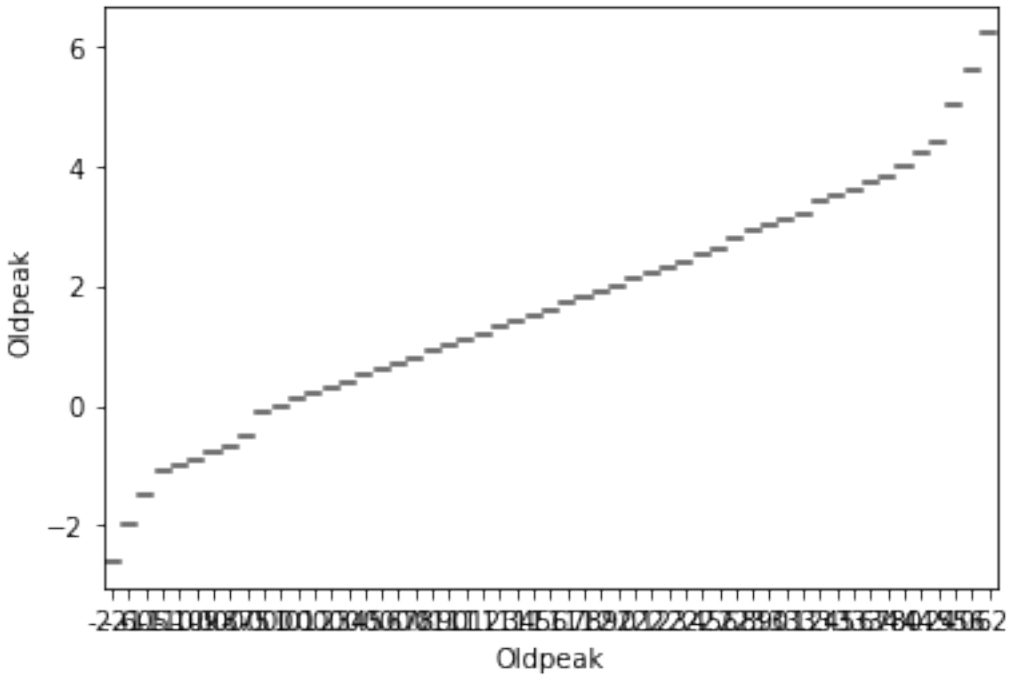
[67]:
```
sns.countplot(x = 'ChestPainType' , hue = 'ChestPainType' ,  data = train_data)
plt.title("Distribution of no. of chest pain w.r.t. Heart Attack" , fontsize=␣
 ↪15)
```

[67]:
```
Text(0.5, 1.0, 'Distribution of no. of chest pain w.r.t. Heart Attack')
```



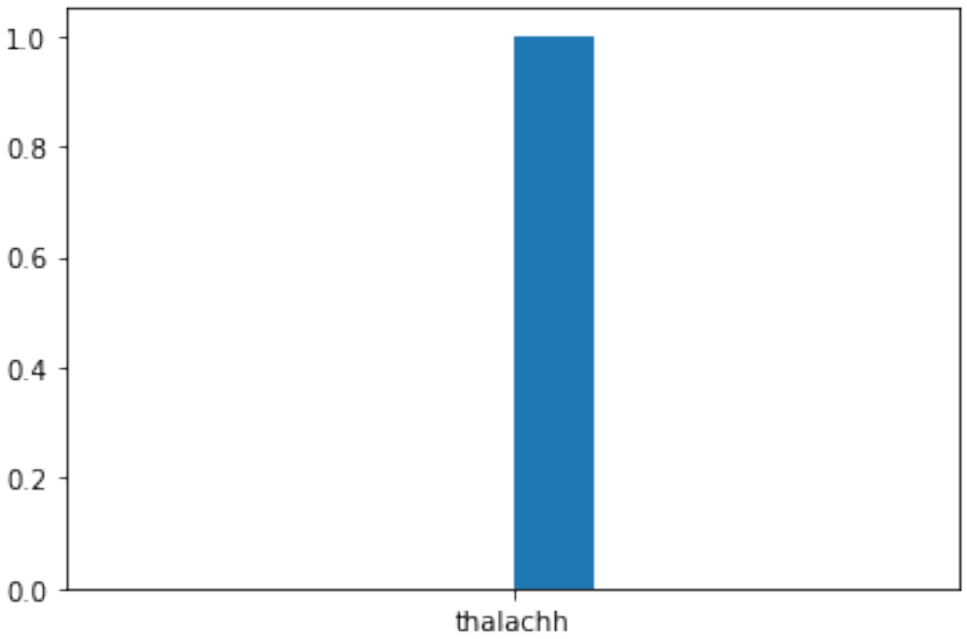
[74]:
```
sns.violinplot(x='Oldpeak',y='Oldpeak',data=train_data)
```

[74]: <AxesSubplot:xlabel='Oldpeak', ylabel='Oldpeak'>



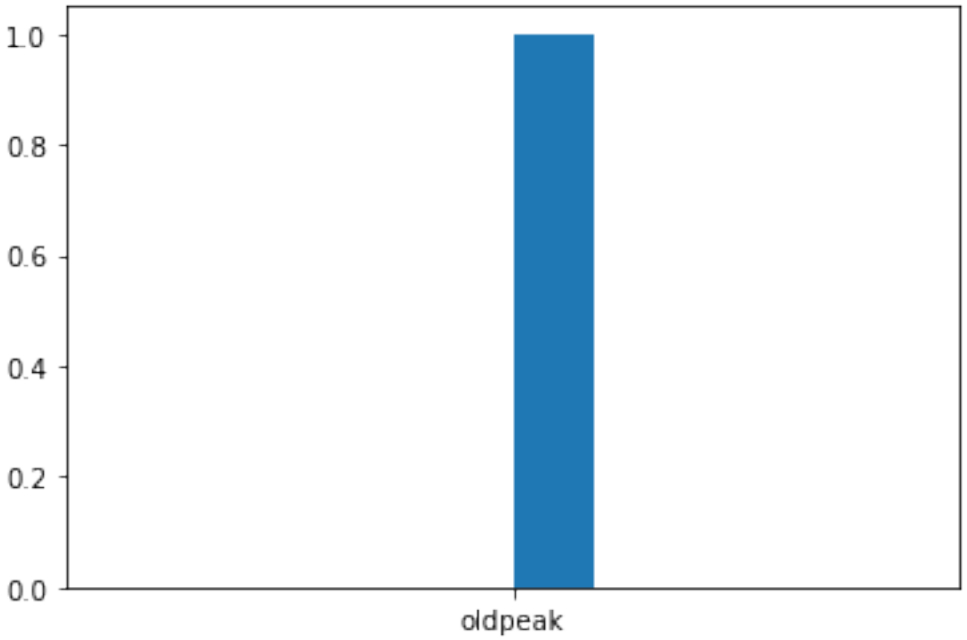
[75]:
```
plt.hist(x="thalachh",data=train_data)
```

[75]:
```
(array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.]),
 array([-0.5, -0.4, -0.3, -0.2, -0.1,  0. ,  0.1,  0.2,  0.3,  0.4,  0.5]),
 <BarContainer object of 10 artists>)
```



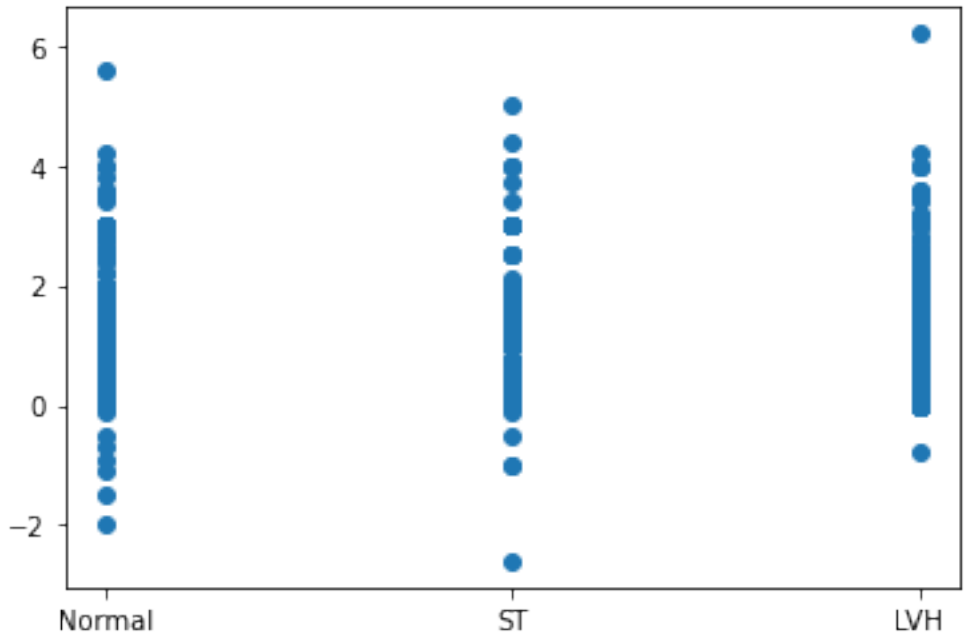
[76]:
```
plt.hist(x="oldpeak",data=train_data)
```

[76]:
```
(array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.]),
 array([-0.5, -0.4, -0.3, -0.2, -0.1,  0. ,  0.1,  0.2,  0.3,  0.4,  0.5]),
 <BarContainer object of 10 artists>)
```

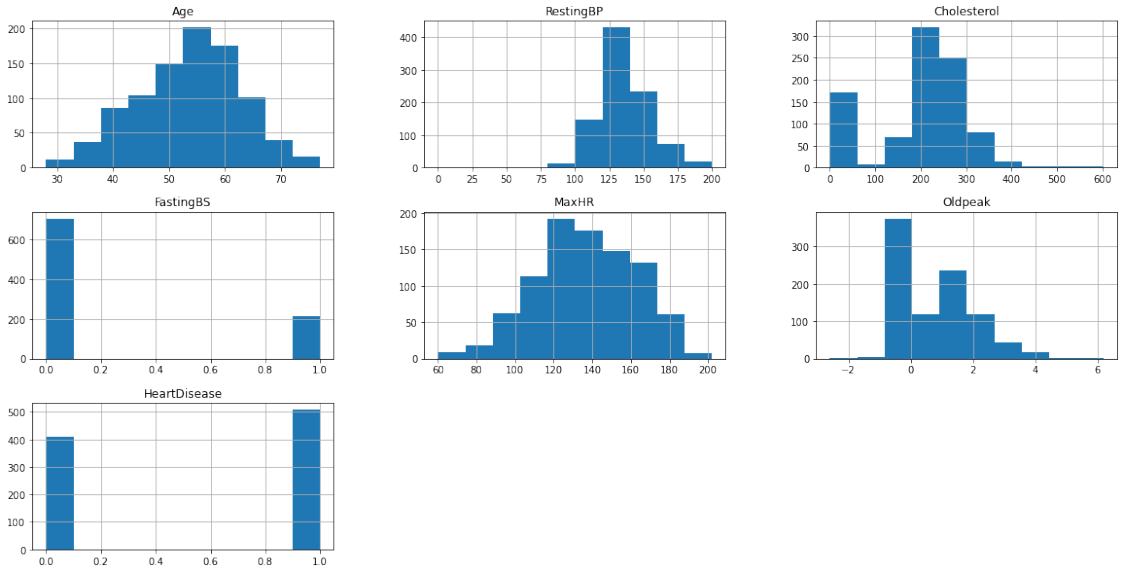


[88]:
```
plt.scatter(train_data['RestingECG'], train_data['Oldpeak'])
```

[88]: <matplotlib.collections.PathCollection at 0x2c4470a59a0>



[93]:
```
train_data.hist(figsize=(20,10))
plt.show()
```



[94]:
```
train_data.corr()
```

[94]:

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	\
Age	1.000000	0.254399	-0.095282	0.198039	-0.382045	0.258612	
RestingBP	0.254399	1.000000	0.100893	0.070193	-0.112135	0.164803	
Cholesterol	-0.095282	0.100893	1.000000	-0.260974	0.235792	0.050148	
FastingBS	0.198039	0.070193	-0.260974	1.000000	-0.131438	0.052698	
MaxHR	-0.382045	-0.112135	0.235792	-0.131438	1.000000	-0.160691	
Oldpeak	0.258612	0.164803	0.050148	0.052698	-0.160691	1.000000	
HeartDisease	0.282039	0.107589	-0.232741	0.267291	-0.400421	0.403951	

	HeartDisease
Age	0.282039
RestingBP	0.107589
Cholesterol	-0.232741
FastingBS	0.267291
MaxHR	-0.400421
Oldpeak	0.403951
HeartDisease	1.000000

[95]:
```
plt.figure(figsize=(30,15))
a = sns.heatmap(train_data.corr(), annot=True, fmt='.1f')
sns.heatmap(train_data.corr())
```

[95]: <AxesSubplot:>

[100]:
```
from sklearn import datasets
```

[102]:
```
X_train = train_data.drop(['Age'],axis =1).values
Y_train = train_data['Age'].values
```

[103]:
```
from sklearn.model_selection import train_test_split
x_train , x_test , y_train , y_test = train_test_split(X_train, Y_train,␣
 ↪test_size = 0.20)
```

[112]:
```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
```